IN THE CLAIMS:

Please amend claims 1, 19, 20, 24, 29-31, 37 and 38 as indicated in the following.

Claims Listing:

1. (Currently Amended) A method comprising:

receiving configuration data that indicates, for each client of a plurality of clients, which ~~one of~~memory controller of a plurality of memory controllers will support ~~each of a plurality of clients~~the client[[,]];

receiving, at a router, one or more data access requests from [[each]]one or more of the plurality of clients; and

routing each of the one or more data access requests from the router to ~~one of~~a corresponding memory controller of the plurality of memory controllers based on the configuration data.

2. (Original) The method as in Claim 1, wherein the plurality of clients and the plurality of memory controllers are integrated on a single device.

3. (Original) The method as in Claim 2, wherein the single device is a semiconductor device.

4. (Original) The method as in Claim 3, wherein the semiconductor device includes a graphics controller.

5. (Original) The method as in Claim 1, wherein data access requests routed to the plurality of memory controllers are executed by the plurality of memory controllers in the order in which the data access requests are received.

6. (Original) The method as in Claim 1, wherein the plurality of clients include at least two clients having a common client type.

7. (Original) The method as in Claim 6, wherein the common client type includes one of the group of a two-dimensional graphics driver, a three dimensional graphics driver, and an audio driver.

8. (Previously Presented) The method as in Claim 6, wherein routing includes routing access requests from a first client having the common client type to a first memory controller and routing access requests from a second client having the common client type to a second memory controller.

9. (Original) The method as in Claim 8, wherein the first memory controller executes a first portion of an access request and the second memory controller executes a second portion of the access request.

10. (Previously Presented) The method as in Claim 6, wherein routing includes routing access requests from a first client having the common client type to a first memory controller and routing access requests from a second client having the common client type to the first memory controller.

11. (Previously Presented) The method as in Claim 6, wherein routing includes routing a first access request from a first client having the common client type to a first memory controller and routing a second access request from a second client having the common client type to the first memory controller, wherein the first and second access requests are simultaneously pending at the first memory controller.

12. (Previously Presented) The method as in Claim 11 further comprising prioritizing one of the first access request and the second access request based upon a predefined arbitration scheme.

13. (Original) The method as in Claim 12, wherein the predefined arbitration scheme is a round robin arbitration scheme.

14. (Original) The method as in Claim 12, wherein the predefined arbitration scheme prioritizes access requests from the first client over access requests from the second client.

15. (Previously Presented) The method as in Claim 11 further comprising prioritizing one of the first access request and the second access request based upon a first predefined arbitration scheme when the first and the second access requests are from clients having a first common client type, and prioritizing one of the first access request and the second access request based upon a second predefined arbitration scheme when the first and the second access requests are from clients having a first common client type.

16. (Original) The method as in Claim 15, wherein a priority assigned to the access requests is dynamic.

17. (Original) The method as in Claim 16, wherein the priority assigned depends on an identifier within the access request.

18. (Original) The method as in Claim 16, wherein the priority assigned depends on an internal timer.

19. (Currently Amended) The method as in Claim 1, wherein ~~client~~ access requests are routed based on one or more of: an address, a client identifier, client tag information, and data size.

20. (Currently Amended) The method as in Claim 1, wherein a number of access requests routed to a memory controller[[,]] from a particular client[[,]] is dependent on [[the]]a data rate of the particular client.

21. (Original) The method as in Claim 1, wherein the plurality of memory controllers are scalable.

22. (Previously Presented) The method as in Claim 1, wherein:

receiving data access requests includes receiving a first HDTV stream and a second HDTV stream; and

routing includes:

routing the first HDTV stream to the first memory controller; and

routing the second HDTV stream to the second memory controller.

23. (Original) The method as in Claim 1, wherein one of the plurality of memory controllers is dedicated for use in handling requests received from a high-data rate client, where the high-data rate client requests a greater amount of data than other clients.

24. (Currently Amended) An apparatus comprising:

a storage module having a memory location and an output port, the memory location to store data;

a plurality of clients, each of the plurality of clients having a data access port;

a router having a plurality of ~~first input~~ first input ports coupled to the data access port of each of the plurality of clients, a second input port coupled to the output port of the storage module, a first plurality of output ports, and a second plurality of output ports, wherein the router is to route data at each one of the plurality of first input ports to a respective output port of the first or second plurality of output ports based upon the data stored in the storage module;

a first memory controller having a plurality of input ports coupled to the first plurality of output ports of the router;

a second memory controller having a plurality of input ports coupled to the second plurality of output ports of the router; and

a first arbiter having a plurality of input ports coupled to the first plurality of output ports of the router, and an output port, wherein the first arbiter selects one data access request on one of the first plurality of input ports to be provided to the output port.

25. (Original) The apparatus of claim 24, further including:

a second arbiter to receive a plurality of client requests from a plurality of clients, to route each of said requests to one of a plurality of memory controllers based on a programmable value; and

a plurality of memory controllers to order client requests, and to deliver said ordered client requests to memory, at least two of the plurality of clients having a common type.

26. (Original) The apparatus as in Claim 24, wherein said first memory controller and said second memory controller include arbiters, said arbiters to order client requests.

27. (Original) The apparatus as in Claim 26, wherein said arbiters perform round robin arbitration between clients having a common type.

28. (Original) The apparatus as in Claim 26, wherein said arbiters performing a first arbitration between clients having a common type, and a second arbitration between clients having different types.

29. (Currently Amended) The apparatus as in Claim 24, wherein said router routes a first client request from a first client to a first memory controller[[,]] and routes a second client request from a second client, the second client being the same client type as the first client, to a second memory controller.

30. (Currently Amended) The apparatus as in Claim 24, wherein said router routes the client requests based on one or more of: an address, a client identifier, client tag information, [[and]]or data size.

31. (Currently Amended) The apparatus as in Claim 24, wherein a number of requests routed to a memory controller[[,]] from a particular client[[,]] is dependent on [[the]]a data rate of the particular client.

32. (Original) The apparatus as in Claim 24, wherein said plurality of memory controllers are scalable.

33. (Original) The apparatus as in Claim 24, wherein:

said router receives a first HDTV stream from a first client and a second HDTV stream from a second client; and wherein

said router routes the first HDTV stream to a first memory controller and routes the second HDTV stream to a second memory controller.

34. (Original) The apparatus as in Claim 24, wherein one of said plurality of memory controllers is dedicated for use in handling requests received from a high-data rate client.

35. (Original) A method comprising:

receiving a first client request from a first video decoder;

routing the first client request to a first memory controller;

receiving a second client request from a second video decoder; and

routing the second client request to a second memory controller.

36. (Original) The method as in Claim 35, further including providing the first client request to a first memory and the second client request to a second memory.

37. (Currently Amended) The method as in Claim 35, wherein routing is based on one or more of: an address, a client identifier, client tag information, [[and]]or data size.

38. (Currently Amended) The method as in Claim 35, wherein a number of requests routed to a memory controller[[,]] from a particular client[[,]] is dependent on the data rate of the particular client.

39. (Previously Presented) The method as in Claim 35, wherein the memory controllers are scalable.

40. (Original) The method as in Claim 35, wherein:
the first video decoder is an MPEG decoder;
the first client request is a first HDTV stream;
the second video decoder is an MPEG decoder; and
the second client request is a second HDTV stream.

41. (Original) The method as in Claim 35, wherein one of the memory controllers is dedicated for use in handling requests received from a high-data rate client.